Assignment 5: Data Visualization

Laila Abed

Fall 2024

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Visualization

Directions

1. Rename this file `<FirstLast>_A05_DataVisualization.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure your code is tidy; use line breaks to ensure your code fits in the knitted output.
5. Be sure to **answer the questions** in this assignment document.
6. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up your session

1. Set up your session. Load the tidyverse, lubridate, here & cowplot packages, and verify your home directory. Read in the NTL-LTER processed data files for nutrients and chemistry/physics for Peter and Paul Lakes (use the tidy `NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv` version in the Processed_KEY folder) and the processed data file for the Niwot Ridge litter dataset (use the `NEON_NIWO_Litter_mass_trap_Processed.csv` version, again from the Processed_KEY folder).

2. Make sure R is reading dates as date format; if not change the format to date.

```
#1
#Loading libraries and reading CVS processed datasets.
library(tidyverse);library(lubridate);library(here)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
## here() starts at /home/guest/EDE_Fall2024
```

```
library(ggridges); library(cowplot); library(ggplot2)
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
library(dplyr)
here()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```
getwd()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```
NTL.LTER.Lake.Chemistry.Nutrients.PeterPaul <-
  read.csv(here(
"./Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"
), stringsAsFactors = TRUE)
NEON.NIWO.Litter.mass.trap <-
  read.csv(here(
"./Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
stringsAsFactors = TRUE)

#2
# Change sampledate and collectdate to dates.
class(NTL.LTER.Lake.Chemistry.Nutrients.PeterPaul$sampledate)
```

```
## [1] "factor"
```

```
class(NEON.NIWO.Litter.mass.trap$collectDate)
```

```
## [1] "factor"
```

```
NTL.LTER.Lake.Chemistry.Nutrients.PeterPaul$sampledate <- as.Date(
  NTL.LTER.Lake.Chemistry.Nutrients.PeterPaul$sampledate, Format="%Y-%m-%d")
NEON.NIWO.Litter.mass.trap$collectDate <- as.Date(
  NEON.NIWO.Litter.mass.trap$collectDate, Format="%Y-%m-%d")
```

Define your theme

3. Build a theme and set it as your default theme. Customize the look of at least two of the following:

- Plot background
- Plot title

- Axis labels
- Axis ticks/gridlines
- Legend

```
#3
#Create a custom theme
my_theme <- function() {
theme_minimal() +
  theme(
# Plot background
plot.background = element_rect(fill = "lightgray"),
# Plot title
plot.title = element_text(color = "navyblue", size = 18, face = "bold"),
# Axis labels
axis.title = element_text(color = "darkgreen", size = 14, face = "italic"),
# Axis gridlines
axis.text = element_text(color = "purple", size = 12),
axis.line = element_line(color = "darkorange"),
axis.ticks = element_line(color = "darkred"),
# Legend
legend.title = element_text(color = "navyblue", size = 12, face = "bold"),
legend.text = element_text(color = "yellow", size = 10)
  )
}
  # Set my theme as the default theme
theme_set(my_theme())
```

Create graphs

For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

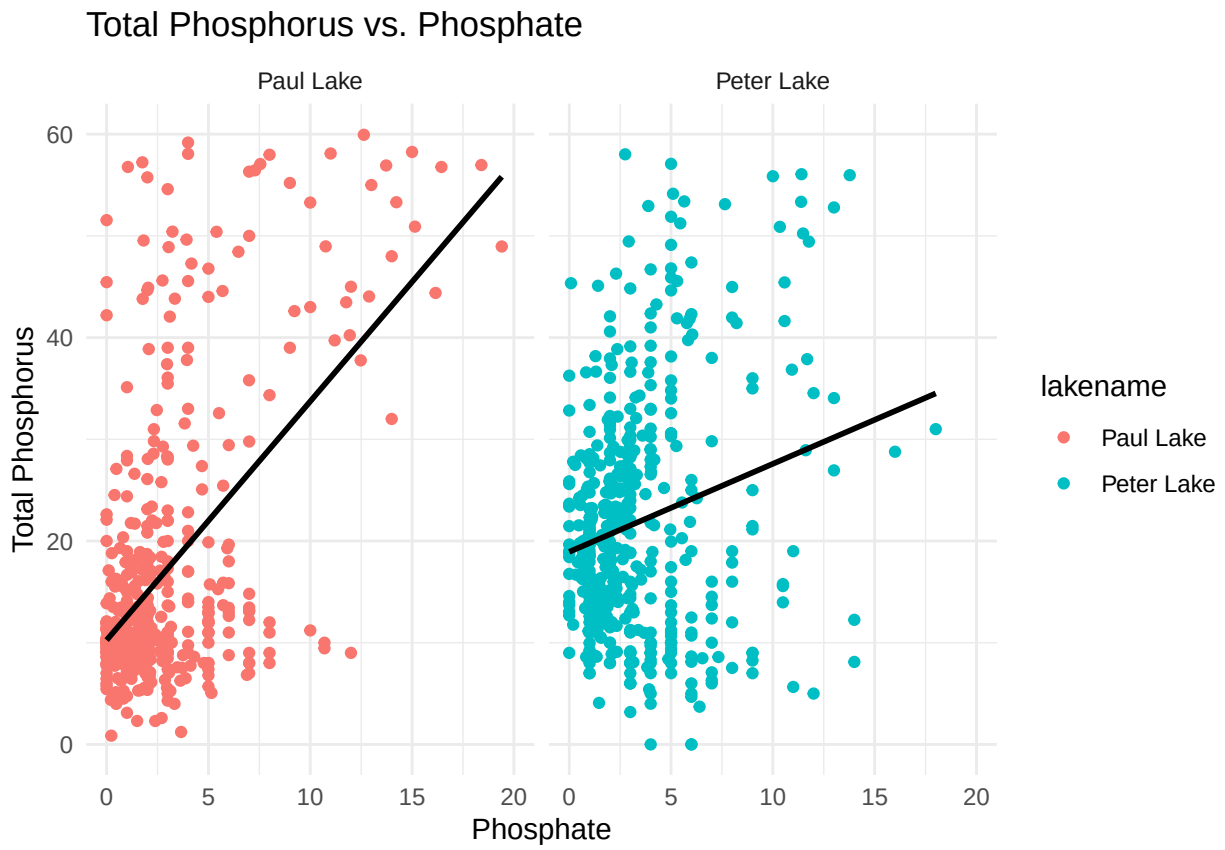
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4
phosphorus_phosphate_plot <- NTL.LTER.Lake.Chemistry.Nutrients.PeterPaul %>%
   ggplot(
    mapping = aes(
      x=po4,
      y=tp_ug,
      color=lakename)
    ) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, color = "black", se = FALSE) +
labs(
title = "Total Phosphorus vs. Phosphate",
x = "Phosphate",
y = "Total Phosphorus"
) +
 xlim(0, 20) +
```

```
  ylim(0, 60) + theme_minimal() +
facet_wrap(~ lakename, ncol = 2) # Separate plots for Peter and Paul lakes
print(phosphorus_phosphate_plot)
```

```
## Warning: Removed 22028 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 22028 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
 ## How we identified 20 and 60 is from the quantile
#scale_y_continuous(limits = c(0,
#quantile(NTL.LTER.Lake.Chemistry.Nutrients.PeterPaul$tp_ug, 0.95,
#         na.rm = TRUE))) +
#scale_x_continuous(limits = c(0,
#quantile(NTL.LTER.Lake.Chemistry.Nutrients.PeterPaul$po4, 0.95,
#         na.rm = TRUE))) +
```

5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

Tips: * Recall the discussion on factors in the lab section as it may be helpful here. * Setting an axis title in your theme to `element_blank()` removes the axis title (useful when multiple, aligned plots use the same

axis values) * Setting a legend's position to "none" will remove the legend from a plot. * Individual plots can have different sizes when combined using `cowplot`.

```
#5
#boxplot temperature
NTL.LTER.Lake.Chemistry.Nutrients.PeterPaul$month <- factor(
NTL.LTER.Lake.Chemistry.Nutrients.PeterPaul$month,
levels=c("1","2","3","4","5","6","7", "8", "9", "10", "11", "12"),
labels=c(
  "Jan","Feb","Mar","Apr","May","Jun","Jul", "Aug", "Sep", "Oct", "Nov","Dec"))

plot_temp <- NTL.LTER.Lake.Chemistry.Nutrients.PeterPaul %>%
  ggplot(
    aes(x=month, y = temperature_C,
        fill = lakename)) +
  geom_boxplot() +
  labs(title = "Boxplot of Temperature", y = "Temperature (°C)") +
  theme_minimal() +
  scale_x_discrete(drop=FALSE )+
  theme(
    axis.title.x = element_blank(),
    legend.position = "none"
  )

#boxplot total phosphorus
plot_tp <- NTL.LTER.Lake.Chemistry.Nutrients.PeterPaul %>%
  ggplot(
    aes(x = month, y = tp_ug, fill = lakename)) +
geom_boxplot() +
labs(title = "Boxplot of Total Phosphorus (TP)", y = "Total P") +
theme_minimal() +
scale_x_discrete(drop=FALSE ) +
  theme(
    axis.title.x = element_blank(),
    legend.position = "none"
  )
#boxplot total nitrogen
plot_tn <- NTL.LTER.Lake.Chemistry.Nutrients.PeterPaul %>%
  ggplot(aes(x =month,
y = tn_ug, fill = lakename)) +
geom_boxplot() +
labs(title = "Boxplot of Total Nitrogen (TN)", y = "Total N") +
theme_minimal() +
scale_x_discrete(drop=FALSE ) +
  theme(
    axis.title.x = element_blank()
  )

## Combine the three plots into one figure with aligned axes

combined_plot <- plot_grid(
  plot_temp + theme(legend.position = "right"), # Remove legend from this plot
plot_tp + theme(legend.position = "right"), # Remove legend from this plot
plot_tn + theme(legend.position = "right"), # Add legend to this plot
```

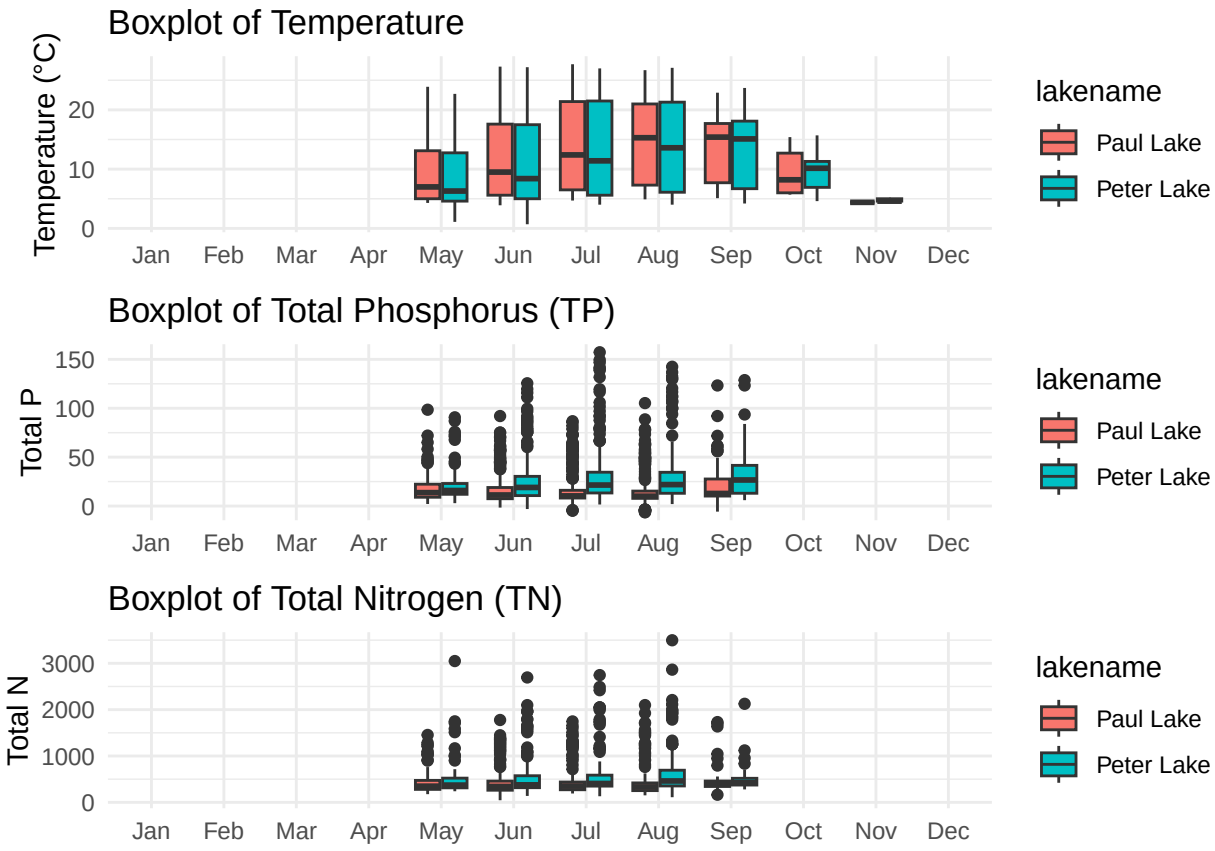
```
ncol = 1, # Stack the plots vertically
align = 'v', # Align vertically
axis = "l", # Align axis labels on the left
rel_heights = c(1, 1, 1)
)
```

```
## Warning: Removed 3566 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

```
## Warning: Removed 20729 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

```
## Warning: Removed 21583 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

```
#Print the combined plot.
print(combined_plot)
```



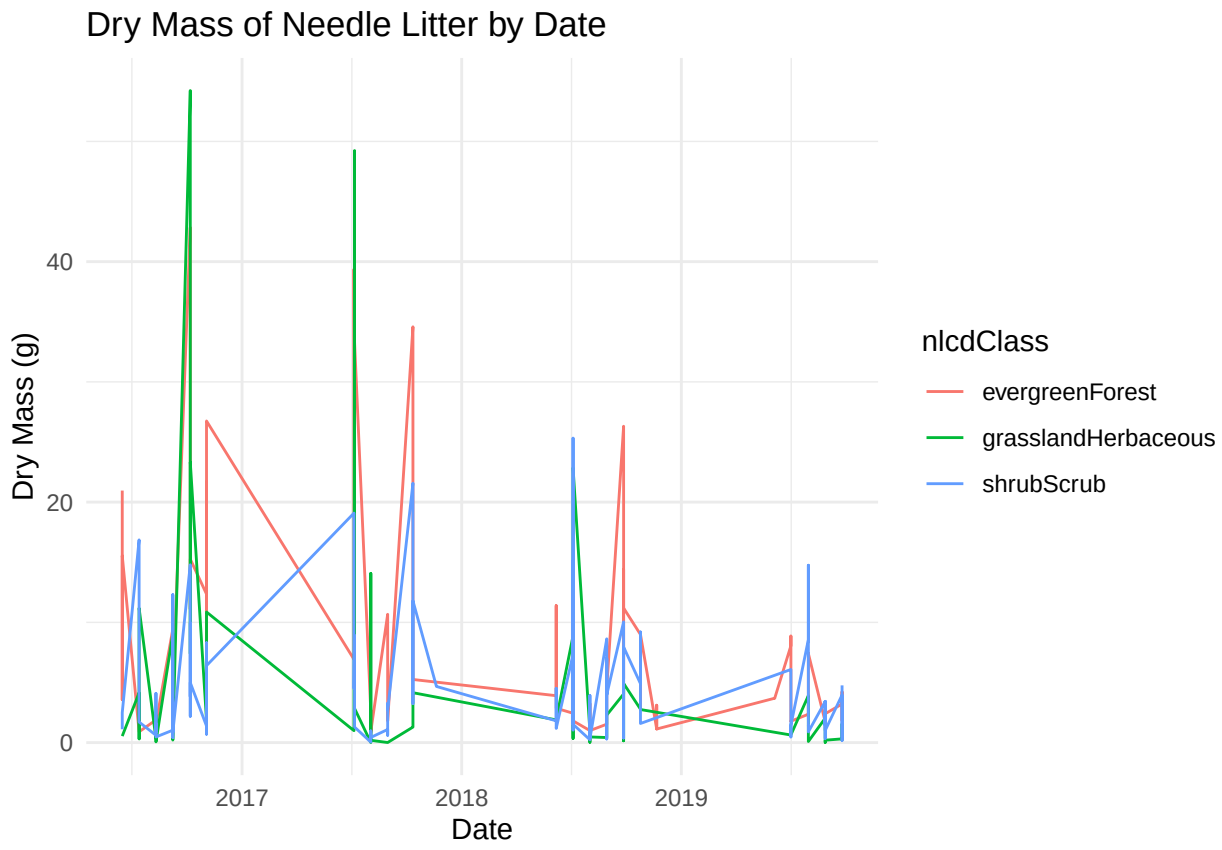
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: From the boxplot of Temperature, Paul Lake has higher temperatures than Peter Lakewe every month except for July. From the boxplot of Total Phosphorus, Paul Lake has lower Phosphorus levels. From the Boxplot of Total Nitrogen, Paul Lake has lower levels of Nitrogen. In general, Paul Lake has higher temperature and lower Phosphorus and Nitrogen levels. I don't understand any thing about environment but from the data, these variables maybe connected or has corrolation or causal relation.

6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

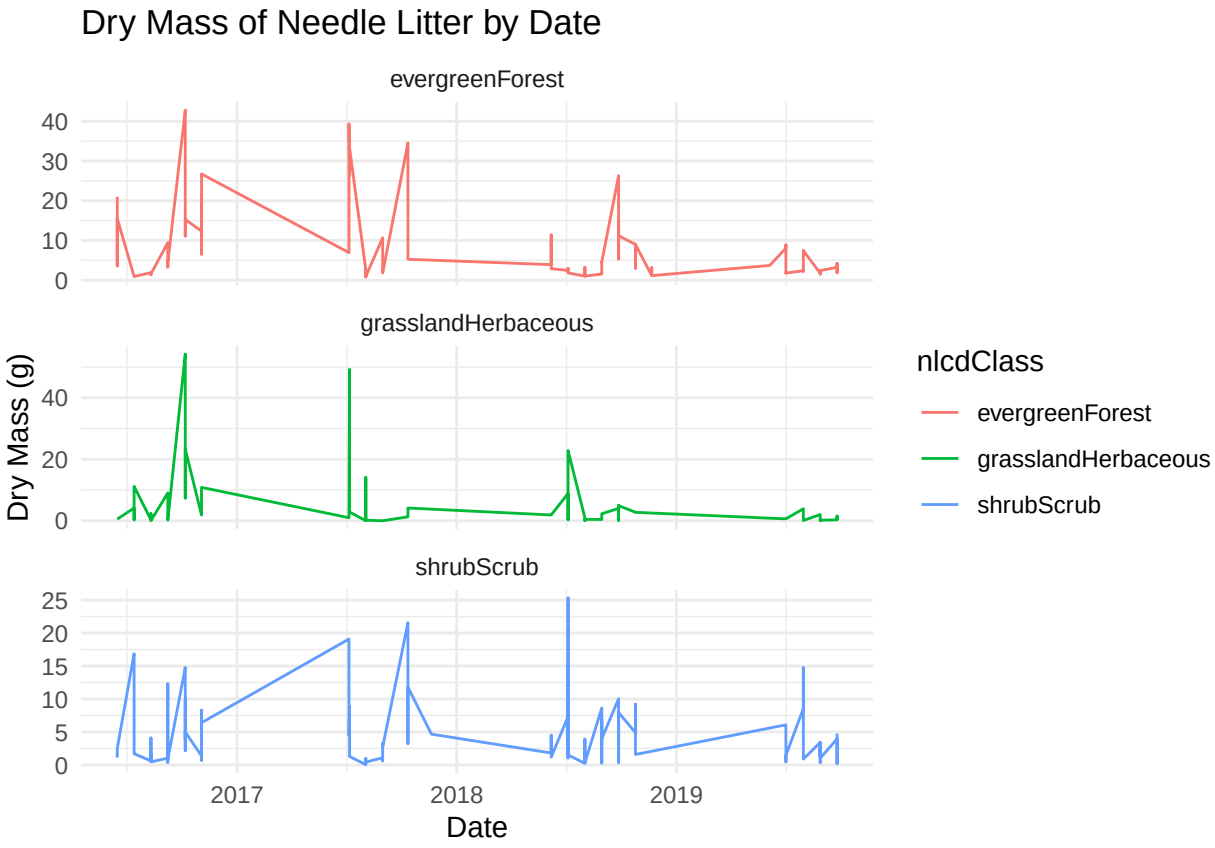
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
##filter the dataset to include only the "Needles" functional group
needles_datasubset <- NEON.NIWO.Litter.mass.trap %>%
filter(functionalGroup == "Needles")
# Now I create the plot
ggplot(needles_datasubset,
       aes(x = collectDate, y = dryMass, color = nlcdClass)) +
geom_line() +
labs(
title = "Dry Mass of Needle Litter by Date",
x = "Date", y = "Dry Mass (g)"
) +
theme_minimal()
```




```
#7
# Use the filtered datasubset that includes only the "Needles" functional group
# needles_datasubset
# Create the plot with facets
ggplot(needles_datasubset, aes(x = collectDate, y = dryMass)) +
```

```
geom_line(aes(color = nlcdClass)) +
labs(
title = "Dry Mass of Needle Litter by Date",
x = "Date",
y = "Dry Mass (g)"
) +
  # Separate by nlcdClass into facets
facet_wrap(~ nlcdClass, scales = "free_y", ncol = 1) +
theme_minimal()
```



Question: Which of these plots (6 vs. 7) do you think is more effective, and why?

Answer: It depends on what we are looking for and what information or question we are answering using these graphs. For example, for data visualisation and for this exercise, Plot 7 is more easier to understand and shows the distribution of the dry mass within different years (peaks and bottoms, stability, etc.) because they are only 3 nlcd classes. However, if we want to compare the datasets between NLCD classes, it would be better to use plot 6 as it shows the trends within and between classes.